Principal Stresses and Mohr's Circle for Plane Stress

Initialization Code (optional)

Manipulate

```
Manipulate[

 Module[{z},
  z = θ * Pi / 180;
  If[plotType == "stress section" || plotType == "Mohr circle/stress section" || plotType == "Mohr circle",
   If[angleSelection == "specific plane",
    z = getAngleAtSpecificPlane[specificPlaneAngle, σx, σy, τxy];
    θ = z * 180. / Pi
   ]
  ];

  Text@makeDiagrams[N@σx, N@σy, N@τxy, z, annotate, onPositiveSideOnly, plotType, limit, gridLines]
 ],

 Grid[{
   {
    Grid[{
      {Style[Row[{"stresses at 0", Degree}], 12], SpanFromLeft},
      {"σx", Control[{{σx, 14, ""}, -20, 20, 0.1, ImageSize → Tiny}],
       Style[Dynamic@padIt1[σx, {3, 1}], 11], Spacer[13]},
      {"σy", Control[{{σy, 4, ""}, -20, 20, 0.1, ImageSize → Tiny}], Style[Dynamic@padIt1[σy, {3, 1}], 11]},
      {"τxy", Control[{{τxy, 10, ""}, -20, 20, 0.1, ImageSize → Tiny}],
       Style[Dynamic@padIt1[τxy, {3, 1}], 11]},
      {Style[Dynamic@Row[{matrix, " = ", TraditionalForm[{
             {padIt1[N[σx], {3, 1}], padIt1[N[τxy], {3, 1}]},
             {padIt1[N[τxy], {3, 1}], padIt1[N[σy], {3, 1}]}
            }]
          }], 11], SpanFromLeft}
     }, Spacings → {.5, .5}, Alignment → Center, Frame → True, FrameStyle -> Directive[Thickness[.005], Gray]]
   },
   {
    Grid[{
      {Style["select plot type", 12], SpanFromLeft},
      {
       PopupMenu[Dynamic[plotType],
        { "stress section" → Style["stress section", 11],
         "Mohr circle" → Style[ "Mohr circle", 11],
         "Mohr circle/stress section" → Style[ "Mohr circle/stress section", 11],
         "normal stress trajectory" → Style[ "normal stress trajectory", 11],
         "shear stress trajectory" → Style["shear stress trajectory", 11],
         "normal/shear trajectory" → Style["normal/shear trajectory", 11]
        }, ImageSize -> All, ContinuousAction -> False], SpanFromLeft
      },
      {Row[{Style["annotate", 12], Spacer[1], Checkbox[Dynamic[annotate],
          Enabled -> Dynamic[plotType == "stress section" || plotType == "Mohr circle/stress section"]]}],
       Row[{Style[Column[{"display stresses on", "positive sides only"}, Alignment → Left], 11], Spacer[1],
         Checkbox[Dynamic[onPositiveSideOnly], Enabled -> Dynamic[
```

```
          plotType == "stress section" || plotType == "Mohr circle/stress section"]]}], SpanFromLeft
    }
   }, Spacings → {.2, .5}, Alignment → Center,
   Frame → True, FrameStyle -> Directive[Thickness[.005], Gray]], SpanFromLeft
 },
 {
  Grid[{
    {Grid[{
       {Row[{Style["rotate to new angle" , 12]}], SpanFromLeft},

       {RadioButtonBar[Dynamic[angleSelection], {"slider" → "", "specific plane" → ""},
         Appearance → "Vertical", Enabled → Dynamic[plotType == "stress section" ||
            plotType == "Mohr circle/stress section" || plotType == "Mohr circle"]],
        Grid[{
          {Row[{Control[{{θ, 45, ""}, -90, 90, 1, ImageSize → Tiny, Enabled →
                 Dynamic[(plotType == "stress section" || plotType == "Mohr circle/stress section" ||
                     plotType == "Mohr circle") && angleSelection == "slider"]}],
             Spacer[4], Style[Row[{Dynamic@padIt2s[θ, 3], Degree}], 11]}]},
          {Row[{
             PopupMenu[Dynamic[specificPlaneAngle ,
                {specificPlaneAngle = #; θ = getAngleAtSpecificPlane[specificPlaneAngle, σx, σy, τxy]} &],
               { "first principal plane" → Style["first principal plane", 11],
                 "second principal plane" → Style["second principal plane", 11],
                 "first maximum shear plane" → Style[ "first shear plane", 11],
                 "second maximum shear plane" → Style[ "second shear plane", 11]
               }, ImageSize -> All, Enabled → Dynamic[(plotType == "stress section" ||
                     plotType == "Mohr circle/stress section") && angleSelection == "specific plane"]
              ]
            }]
          }
         }
        ]
       }
      }, Spacings → {.5, .5}, Alignment → Center, Frame → None]
     , Spacer[7]
    }
    ,
    {Style[Dynamic@Row[{matrix2, " = ",
         TraditionalForm[{
           {padIt1[N[1/2 (σx + σy) + 1/2 (σx - σy) Cos[2 (θ * Pi / 180)] + τxy Sin[2 (θ * Pi / 180)]], {3, 1}],
            padIt1[N[-1/2 (σx - σy) Sin[2 (θ * Pi / 180)] + τxy Cos[2 (θ * Pi / 180)]], {3, 1}]
           },
           {padIt1[N[-1/2 (σx - σy) Sin[2 (θ * Pi / 180)] + τxy Cos[2 (θ * Pi / 180)]], {3, 1}],
            padIt1[N[1/2 (σx + σy) - 1/2 (σx - σy) Cos[2 (θ * Pi / 180)] - τxy Sin[2 (θ * Pi / 180)]], {3, 1}]
           }
          }
         ]}], 11]
    }}, Spacings → {.5, .5}, Alignment → Center,
   Frame → True, FrameStyle -> Directive[Thickness[.005], Gray]]
 },
 {
  Grid[{
```

```
      {Grid[{
          {Style["zoom", 12], Spacer[13],
           Control[{{limit, 30, ""}, 5, 50, 0.1, ImageSize → Small}], Spacer[12]},
          {"", Style[Row[{"in", Spacer[75], "out"}], 11], SpanFromLeft}
         }, Spacings → {.1, .1},
         Alignment → Center, Frame → True, FrameStyle -> Directive[Thickness[.005], Gray]]
      },
      {Grid[{
          {Style["gridlines", 12], Control[{{gridLines, 0.5, ""}, 0, 1, 0.1, ImageSize → Small}] Spacer[8]},
          {"", Style[Row[{"less", Spacer[70], "more"}], 11]}
         }, Spacings → {.1, .1},
         Alignment → Center, Frame → True, FrameStyle -> Directive[Thickness[.005], Gray]]
      }
     }, Spacings → {.1, .5}, Alignment → Left, Frame → None
    ], SpanFromLeft
   }

  }, Spacings → {.2, .5}, Alignment → Left],

 {{annotate, True}, None},
 {{plotType, "Mohr circle"}, None},

 {{matrix, TraditionalForm[{{σx, τxy}, {τxy, σy}}]}, None},
 {{matrix2, TraditionalForm[{{(σ')x, τ'xy}, {τ'xy, (σ')y}}]}, None},

 {{onPositiveSideOnly, True}, None},
 {{principalPlaneAngle, False}, None},
 {{maxShearPlane, False}, None},
 {{specificPlaneAngle, "first principal plane"}, None},
 {{angleSelection, "slider"}, None},

 ControlPlacement → Left,
 SynchronousUpdating → False,
 SynchronousInitialization → False,
 ContinuousAction → True,
 Alignment -> Center,
 ImageMargins → 0,
 FrameMargins → 0,
 Paneled → True,
 Frame → False,
 AutorunSequencing → {1},
 Initialization :→
  {
   (*--- constant parameters size and width of display ---*)
   contentSizeW = 425;
   contentSizeH = 425;

   (*--------------------------------------------*)
   (* helper function for formatting             *)
   (*--------------------------------------------*)
   padIt2[v_?numeric, f_List] :=
    AccountingForm[Chop[v] , f, NumberSigns → {"", ""}, NumberPadding → {"0", "0"}, SignPadding → True];
   padIt2[v_?numeric, f_Integer] := AccountingForm[Chop[v] , f, NumberSigns → {"", ""},
     NumberPadding → {"0", "0"}, SignPadding → True];
   padIt2s[v_?numeric, f_Integer] := AccountingForm[Chop[v] , f, NumberSigns → {"-", "+"},
     NumberPadding → {"0", "0"}, SignPadding → True];
   padIt1[v_?numeric, f_List] := AccountingForm[Chop[v] , f, NumberSigns → {"-", "+"},
     NumberPadding → {"0", "0"}, SignPadding → True];

   (*definitions used for parameter checking*)
   integerStrictPositive = (IntegerQ[#] && # > 0 &);
   integerPositive = (IntegerQ[#] && # ≥ 0 &);
```

```
numericStrictPositive = (Element[#, Reals] && # > 0 &);
numericPositive = (Element[#, Reals] && # ≥ 0 &);
numericStrictNegative = (Element[#, Reals] && # < 0 &);
numericNegative = (Element[#, Reals] && # ≤ 0 &);
bool = (Element[#, Booleans] &);
numeric = (Element[#, Reals] &);
integer = (Element[#, Integers] &);

(*-------------------------------------------------------------*)
makeDiagrams[σx_?numeric, σy_?numeric, τxy_?numeric, θ_?numeric, annotate_?bool, onPositiveSideOnly_?
   bool, plotType_String, limit_?numericStrictPositive, gridLines_?numericPositive] := Module[{},

  Which[

   plotType == "stress section", make2DStressDiagram[σx, σy, τxy, θ,
    annotate, onPositiveSideOnly, limit, gridLines, {contentSizeW, contentSizeH}],

   plotType == "Mohr circle", makeMohrCircle[θ, σx, σy, τxy, limit,
    gridLines, {contentSizeW, contentSizeH}, makeMohrCircleTitle[σx, σy, τxy]],

   plotType == "Mohr circle/stress section",
   Grid[{
     {makeMohrCircleTitle[σx, σy, τxy], SpanFromLeft},
     {make2DStressDiagram[σx, σy, τxy, θ, annotate,
       onPositiveSideOnly, limit, gridLines, {0.5 contentSizeW, 0.87 contentSizeH}],
      makeMohrCircle[θ, σx, σy, τxy, limit, gridLines, {0.499 contentSizeW, .87 contentSizeH}, {}]
     }
    }, Spacings → {0, 0}
   ],

   plotType == "normal stress trajectory", makeNormalStressPolarPlot[σx, σy, τxy, limit, gridLines],

   plotType == "shear stress trajectory", makeShearStressPolarPlot[σx, σy, τxy, limit, gridLines],

   plotType == "normal/shear trajectory",
   makeShearAndNormalStressPolarPlot[σx, σy, τxy, limit, gridLines]
  ]
 ];

(*-------------------------------------------------------------*)
getAngleAtSpecificPlane[specificPlaneAngle_, σx_, σy_, τxy_] :=
 N@Which[specificPlaneAngle == "first principal plane", principalStresses[σx, σy, τxy][[1, 2]],
   specificPlaneAngle == "second principal plane", principalStresses[σx, σy, τxy][[2, 2]],
   specificPlaneAngle == "first maximum shear plane", principalStresses[σx, σy, τxy][[1, 2]] + Pi / 4,
   specificPlaneAngle == "second maximum shear plane", principalStresses[σx, σy, τxy][[1, 2]] + 3/4 Pi
  ];
(*-------------------------------------------------------------*)
(*finds the 2 Principal stresses in plane stress 2D setting*)
principalStresses[σx_?numeric, σy_?numeric, τxy_?numeric] :=
 Module[{θp, σ, σ1, σ2, σ1max, r, c, tmp, θ1, θ2},

  r = Sqrt[((σx - σy)/2)^2 + τxy^2];
  c = (σx + σy)/2;
  {σ1, σ2} = {c + r, c - r};

  (*σ1 is the largest stress regadless of sign*)
  If[Abs[σ2] > Abs[σ1],
   tmp = σ1;
```

```
    σ1 = σ2;
    σ2 = tmp
   ];

   If[Abs[σx - σy] ≤ $MachineEpsilon, θp = Pi / 4, θp = ArcTan[2 Abs[τxy] / Abs[σx-σy]] / 2];

   If[σ1 > σ2,
    If[τxy > 0, (*below*)
     If[σx > c, {θ1, θ2} = {θp, -(Pi / 2 - θp)}, {θ1, θ2} = {Pi / 2 - θp, -θp}]
     ,
     If[σx > c, {θ1, θ2} = {-θp, (Pi / 2 - θp)}, {θ1, θ2} = {-(Pi / 2 - θp), θp}]
    ]
    ,
    If[τxy > 0,
     If[σx > c, {θ1, θ2} = {-(Pi / 2 - θp), θp}, {θ1, θ2} = {-θp, Pi / 2 - θp}]
     ,
     If[σx > c, {θ1, θ2} = {(Pi / 2 - θp), -θp}, {θ1, θ2} = {θp, -(Pi / 2 - θp)}]
    ]
   ];

   {{σ1, θ1}, {σ2, θ2}}
  ];
(*-------------------------------------------------------------*)
(*finds the maximum and minumum shear stresses in plane stress 2D setting*)
maxAndMinShearStress[σx_?numeric, σy_?numeric, τxy_?numeric] := Module[{r},

   r = Sqrt[((σx - σy)/2)^2 + τxy^2];
   {r, -r}
  ];
(*-------------------------------------------------------------*)
(*find normal and shear stress for plane at angle theta from normal. plain stress*)
(*use standard stress angle transformation for 2D*)
rotationStress[σx_, σy_, τxy_, θ_] := Module[{σxx, σyy, τ},

   σxx = 1/2 (σx + σy) + 1/2 (σx - σy) Cos[2 θ] + τxy Sin[2 θ];

   τ = -1/2 (σx - σy) Sin[2 θ] + τxy Cos[2 θ];

   σyy = 1/2 (σx + σy) - 1/2 (σx - σy) Cos[2 θ] - τxy Sin[2 θ];

   {σxx, σyy, τ}
  ];
(*-------------------------------------------------------------*)
plot[data_List, limit_?numericStrictPositive, gridLines_?numericPositive, color_] :=

 ListPolarPlot[data,
  Joined → True,
  AxesOrigin → {0, 0},
  ImageSize → {contentSizeW, contentSizeH},
  ImagePadding → {{20, 10}, {20, 5}},
  ImageMargins → 0,
  AspectRatio → 1,
  Frame → True,
  If[gridLines == 0, GridLines → None, {GridLines →
     {Range[-limit, limit, (2 * limit) / (gridLines * 20)], Range[-limit, limit, (2 * limit) / (gridLines * 20)]},
    GridLinesStyle → Directive[Thickness[.001], LightGray]
   }],
  PlotRange → {{-limit, limit}, {-limit, limit}},
  PlotStyle → color
```

```
 ];
(*-------------------------------------------------------------*)
makeArrowForAngle[r_, center_, {{σ1_, θ1_}, {σ2_, θ2_}}, τxy_?numeric] := Module[{phi, tbl, align},

  If[σ1 > σ2,
   If[τxy > 0,
    tbl = Table[{center[[1]] + r / 3 * Cos[phi], r / 3 * Sin[phi]}, {phi, -2 * θ1, 0, Pi / 100}];
    align = {-1, 1}
    ,
    tbl = Table[{center[[1]] + r / 3 * Cos[phi], r / 3 * Sin[phi]}, {phi, -2 * θ1, 0, -Pi / 100}];
    align = {-1, -1}
   ]
   ,
   If[τxy > 0,
    tbl = Table[{center[[1]] + r / 3 * Cos[phi], r / 3 * Sin[phi]}, {phi, -(Pi + 2 * θ1), -Pi, -Pi / 100}];
    align = {1, 1}
    ,
    tbl = Table[{center[[1]] + r / 3 * Cos[phi], r / 3 * Sin[phi]}, {phi, (Pi - 2 * θ1), Pi, Pi / 100}];
    align = {1, -1}
   ]
  ];

  {Text["2θ1", If[Length[tbl] > 1, tbl[[ Round[ Length[tbl] / 2 ] ]], First@tbl] , align ], tbl}
 ];
(*-------------------------------------------------------------*)
makeMohrCircleTitle[σx_?numeric, σy_?numeric, τxy_?numeric] := Module[{σ1, σ2, θ1, θ2, r, center, ptA},

  {{σ1, θ1}, {σ2, θ2}} = principalStresses[σx, σy, τxy];
  center = {(σ1 + σ2)/2, 0};
  ptA = {σx, -τxy};
  r = EuclideanDistance[center, ptA];

  Grid[{
    TraditionalForm[Style[#]] & /@ {"σx", "σy", "τxy", "σ1", "σ2", "τmax", "θ1", "θ2"},
    {padIt1[σx, {4, 1}],
     padIt1[σy, {4, 1}],
     padIt1[τxy, {4, 1}],
     padIt1[σ1, {4, 1}],
     padIt1[σ2, {4, 1}],
     ±padIt2[r, {4, 1}],
     Row[{padIt1[θ1 * 180 / Pi, {4, 1}], Degree}],
     Row[{padIt1[θ2 * 180 / Pi, {4, 1}], Degree}]
    }
   }, Spacings → {.5, 1}, Frame → All, FrameStyle -> Directive[Thin]]

 ];

(*-------------------------------------------------------------*)
getRadiusOfCircle[θ_?numeric, σx_?numeric, σy_?numeric, τxy_?numeric] := Sqrt[((σx - σy)/2)^2 + τxy^2];

(*-------------------------------------------------------------*)
getCurrentStressOnInclinded[θ_?numeric,
  σx_?numeric, σy_?numeric, τxy_?numeric] := Module[{σx1, τxy1, σy1},
  σx1 = (σx + σy)/2 + ((σx - σy)/2 Cos[2 θ] + τxy Sin[2 θ]);
  τxy1 = (-(σx - σy)/2 Sin[2 θ] + τxy Cos[2 θ]);
  σy1 = (σx + σy)/2 - ((σx - σy)/2 Cos[2 θ] + τxy Sin[2 θ]);
```

```
  {σx1, σy1, τxy1}

 ];

(*-------------------------------------------------------------*)
makeMohrCircle[θ_?numeric, σx_?numeric, σy_?numeric,
  τxy_?numeric, limit_?numericStrictPositive, gridLines_?numericPositive,
  {contentSizeW_?numericStrictPositive, contentSizeH_?numericStrictPositive}, plotTitle_] :=
 Module[{ptA, ptB, center, σ1, σ2, θ1, θ2, r, z, lst, txt, ptD1, ptD2, σx1, τxy1, σy1},


  {{σ1, θ1}, {σ2, θ2}} = principalStresses[σx, σy, τxy];
  center = {(σ1 + σ2)/2, 0};
  ptA = {σx, -τxy};
  r = getRadiusOfCircle[θ, σx, σy, τxy];
  {σx1, σy1, τxy1} = getCurrentStressOnInclinded[θ, σx, σy, τxy];

  (*Print["in makeMohrCircle, θ=",θ," σx=",σx," σy=",σy," r=",r," cosBeta=",cosBeta,
    " sinBeta=",sinBeta," currentStress=",currentStress," currentShear=",currentShear];*)

  z = σx - First@center;
  ptB = {ptA[[1]] - 2 z, -ptA[[2]]};
  ptD1 = {σx1, -τxy1};
  ptD2 = {σy1, τxy1};

  Graphics[{
    Circle[center, r],

    (*{Text[TraditionalForm[Style["(σx,-τxy)",12]],ptA,If[σx>center[[1]],{-1,1},{1,1}]]},*)
    {Black, PointSize[.02], Point[ptA]},
    {Black, PointSize[.02], Point[center]},

    (*{Text[TraditionalForm[Style["(σy,τxy)",12]],ptB,If[σy>center[[1]],{-1,-1},{1,-1}]]},*)
    {Black, PointSize[.02], Point[ptB]},
    {Dashed, Line[{ptA, ptB}]},

    Circle[ptD1, .8],
    {Red, Dashed, Line[{ptD1, ptD2}]},
    Circle[ptD2, .8],
    Text[Row[{"(", padIt1[σx1, {4, 1}], ",", padIt1[τxy1, {4, 1}], ")"}],
     ptD1, If[σx1 > center[[1]], {-1, -1}, {1, -1}]],
    Text[Row[{"(", padIt1[σy1, {4, 1}], ",", padIt1[τxy1, {4, 1}], ")"}],
     ptD2, If[σy1 > center[[1]], {-1, -1}, {1, -1}]],

    {Red, PointSize[.02], Point[{σ1, 0}]},
    {Text[TraditionalForm[Style["σ1", 12]], {σ1, 0}, {-1.5, 1.5}]},

    {Red, PointSize[.02], Point[{σ2, 0}]},
    {Text[TraditionalForm[Style["σ2", 12]], {σ2, 0}, {1.2, 1.3}]},

    {Blue, PointSize[.02], Point[{center[[1]], r}]},
    {Text[TraditionalForm[Style["τmax", 12]], {center[[1]], r}, {0, -1.5}]},

    {Blue, PointSize[.02], Point[{center[[1]], -r}]},
    {Text[TraditionalForm[Style["τmax", 12]], {center[[1]], -r}, {0, 1.5}]},

    {Text[Style["tension", 11], {limit, 0}, {1, 3}]},
    {Text[Style["compression", 11], {-limit, 0}, {-1, 3}]}
   },
   If[gridLines == 0, GridLines → None,
    {GridLines → {Range[-limit, limit, (2 * limit) / (gridLines * 20)], Range[-limit, limit,
```

```
        (2*limit) / (gridLines*20)]}, GridLinesStyle → Directive[Thickness[.001], LightGray]
     }],
    PlotRange → {{-limit, limit}, {-limit, limit}},
    Axes → True,
    AxesOrigin → {0, 0},
    TicksStyle → 8,
    PlotLabel -> If[plotTitle === {}, "", plotTitle],
    ImageSize → {contentSizeW, contentSizeH},
    ImagePadding → {{20, 10}, {20, 5}}
   ]

  ];
(*--------------------------------------------------------------*)
makeShearAndNormalStressPolarPlot[σx_?numeric, σy_?numeric,
   τxy_?numeric, limit_?numericStrictPositive, gridLines_?numericPositive] :=
 Module[{pts, θ, σ1, σ2, σ1Abs, σ2Abs, θp1, θp2, p1, p2, p3, p4, plotTitle, coord1, coord2, τ1, τ2},

  pts = Table[{rotationStress[σx, σy, τxy, θ], θ}, {θ, 0, 2 Pi, Pi / 40}];
  {{σ1, θp1}, {σ2, θp2}} = principalStresses[σx, σy, τxy];
  p1 = plot[Transpose[{pts[[All, 2]], pts[[All, 1, 1]]}], limit, gridLines, Red];

  coord1 = {Abs[σ1] Cos[θp1], Abs[σ1] Sin[θp1]};
  coord2 = {Abs[σ2] Cos[θp2], Abs[σ2] Sin[θp2]};

  p2 = Graphics[{
      {PointSize[0.015], Point[{σx, 0}]},
      Text[TraditionalForm[Style["σx", 12]], {σx, 0}, {0, 1.2}],
      {PointSize[0.015], Point[{0, σy}]},
      Text[TraditionalForm[Style["σy", 12]], {0, σy}, {1.2, 0}],

      {PointSize[0.015], Point[coord1]},
      Text[TraditionalForm[Style["σ1", 12]], coord1, {-1.4, 0}],
      {PointSize[0.015], Point[coord2]},
      Text[TraditionalForm[Style["σ2", 12]], coord2, {-1.4, 0}],
      {Dashed, Thin, Line[{coord1, {-coord1[[1]], -coord1[[2]]}}]},
      {Dashed, Thin, Line[{coord2, {-coord2[[1]], -coord2[[2]]}}]}
     }
    ];

  {τ1, τ2} = maxAndMinShearStress[σx, σy, τxy];
  p3 = plot[Transpose[{pts[[All, 2]], pts[[All, 1, 3]]}], limit, gridLines, Blue];

  coord1 = {Abs[τ1] Cos[θp1 + Pi / 4], Abs[τ1] Sin[θp1 + Pi / 4]};
  coord2 = {Abs[τ1] Cos[θp2 + Pi / 4], Abs[τ1] Sin[θp2 + Pi / 4]};

  p4 = Graphics[{
      {PointSize[0.015], Point[{τxy, 0}]},
      Text[TraditionalForm[Style["τxy", 12]], {τxy, 0}, {0, 1.2}],

      {PointSize[0.015], Point[{0, τxy}]},
      Text[TraditionalForm[Style["τyx", 12]], {0, τxy}, {1.2, 0}],

      {PointSize[0.015], Point[coord1]},
      Text[TraditionalForm[Style["τmax", 12]], coord1, {-1.4, 0}],

      {Dashed, Thin, Line[{coord1, {-coord1[[1]], -coord1[[2]]}}]},
      {Dashed, Thin, Line[{coord2, {-coord2[[1]], -coord2[[2]]}}]}
     }
    ];

  plotTitle = Style[Grid[{
       {"normal (red) and shear (blue) polar (stress vs. angle) trajectory", SpanFromLeft},
       TraditionalForm[Style[#]] & /@ {"σx", "σy", "τxy", "σ1", "θ1", "σ2", "θ2", "τmax"},
```

```
        {padIt1[σx, {4, 1}],
         padIt1[σy, {4, 1}],
         padIt1[τxy, {4, 1}],
         padIt1[σ1, {4, 1}],
         Row[{padIt1[θp1 * 180 / Pi, {4, 1}], Degree}],
         padIt1[σ2, {4, 1}],
         Row[{padIt1[θp2 * 180 / Pi, {4, 1}], Degree}],
         ±padIt2[τ1, {4, 1}]
        }
      }, Spacings → {0.4, 1.1}, Frame → All, FrameStyle -> Directive[Thin]], 12];

   Show[p1, p2, p3, p4, PlotLabel -> plotTitle]

  ];
(*--------------------------------------------------------------*)
makeNormalStressPolarPlot[σx_?numeric, σy_?numeric,
   τxy_?numeric, limit_?numericStrictPositive, gridLines_?numericPositive] :=
 Module[{pts, θ, σ1, σ2, σ1Abs, σ2Abs, θp1, θp2, p1, p2, plotTitle, coord1, coord2},

   pts = Table[{rotationStress[σx, σy, τxy, θ], θ}, {θ, 0, 2 Pi, Pi / 40}];
   {{σ1, θp1}, {σ2, θp2}} = principalStresses[σx, σy, τxy];

   p1 = plot[Transpose[{pts[[All, 2]], pts[[All, 1, 1]]}], limit, gridLines, Red];
   coord1 = {Abs[σ1] Cos[θp1], Abs[σ1] Sin[θp1]};
   coord2 = {Abs[σ2] Cos[θp2], Abs[σ2] Sin[θp2]};

   p2 = Graphics[{
       {PointSize[0.015], Point[{σx, 0}]},
       Text[TraditionalForm[Style["σx", 12]], {σx, 0}, {0, 1.2}],
       {PointSize[0.015], Point[{0, σy}]},
       Text[TraditionalForm[Style["σy", 12]], {0, σy}, {1.2, 0}],

       {PointSize[0.015], Point[coord1]},
       Text[TraditionalForm[Style["σ1", 12]], coord1, {-1.4, 0}],
       {PointSize[0.015], Point[coord2]},
       Text[TraditionalForm[Style["σ2", 12]], coord2, {-1.4, 0}],
       {Dashed, Thin, Line[{coord1, {-coord1[[1]], -coord1[[2]]}}]},
       {Dashed, Thin, Line[{coord2, {-coord2[[1]], -coord2[[2]]}}]}
      }
     ];

   plotTitle = Grid[{
       {"normal stress polar (stress vs. angle) trajectory", SpanFromLeft},
       TraditionalForm[Style[#]] & /@ {"σx", "σy", "τxy", "σ1", "θ1", "σ2", "θ2"},
       {padIt1[σx, {4, 1}],
        padIt1[σy, {4, 1}],
        padIt1[τxy, {4, 1}],
        padIt1[σ1, {4, 1}],
        Row[{padIt1[θp1 * 180 / Pi, {4, 1}], Degree}],
        padIt1[σ2, {4, 1}],
        Row[{padIt1[θp2 * 180 / Pi, {4, 1}], Degree}]
       }
     }, Spacings → {0.8, 1}, Frame → All, FrameStyle -> Directive[Thin]];
   Show[p1, p2, PlotLabel -> plotTitle]
  ];
(*--------------------------------------------------------------*)
makeShearStressPolarPlot[σx_?numeric, σy_?numeric,
   τxy_?numeric, limit_?numericStrictPositive, gridLines_?numericPositive] :=
 Module[{pts, θ, σ1, τ1, τ2, σ2, σ1Abs, σ2Abs, θp1, θp2, p1, p2, plotTitle, coord1, coord2},

   pts = Table[{rotationStress[σx, σy, τxy, θ], θ}, {θ, 0, 2 Pi, Pi / 40}];
   {{σ1, θp1}, {σ2, θp2}} = principalStresses[σx, σy, τxy];
   {τ1, τ2} = maxAndMinShearStress[σx, σy, τxy];
```

```
  p1 = plot[Transpose[{pts[[All, 2]], pts[[All, 1, 3]]}], limit, gridLines, Blue];

  coord1 = {Abs[τ1] Cos[θp1 + Pi / 4], Abs[τ1] Sin[θp1 + Pi / 4]};
  coord2 = {Abs[τ1] Cos[θp2 + Pi / 4], Abs[τ1] Sin[θp2 + Pi / 4]};

  p2 = Graphics[{
     {PointSize[0.015], Point[{τxy, 0}]},
     Text[TraditionalForm[Style["τxy", 12]], {τxy, 0}, {0, 1.2}],

     {PointSize[0.015], Point[{0, τxy}]},
     Text[TraditionalForm[Style["τyx", 12]], {0, τxy}, {1.2, 0}],

     {PointSize[0.015], Point[coord1]},
     Text[TraditionalForm[Style["τmax", 12]], coord1, {-1.4, 0}],

     {Dashed, Thin, Line[{coord1, {-coord1[[1]], -coord1[[2]]}}]},
     {Dashed, Thin, Line[{coord2, {-coord2[[1]], -coord2[[2]]}}]}
    }
   ];

  plotTitle = Grid[{
     {"shear stress polar (stress vs. angle) trajectory", SpanFromLeft},
     Flatten@{TraditionalForm[Style[#]] & /@ {"σx", "σy", "τxy", "τmax", "θmax"}, SpanFromLeft},
     {padIt1[σx, {4, 1}],
      padIt1[σy, {4, 1}],
      ±padIt2[τxy, {4, 1}],
      padIt1[τ1, {4, 1}],
      Row[{padIt1[(θp1 + Pi / 4) * 180 / Pi, {4, 1}],
        Degree, ",", padIt1[(θp1 + 3/4 Pi) * 180 / Pi, {4, 1}], Degree}], SpanFromLeft}
    }, Spacings → {1, 1}, Frame → All, FrameStyle -> Directive[Thin]];
  Show[p1, p2, PlotLabel -> plotTitle]
 ];
(*-------------------------------------------------------------*)
make2DStressDiagram[σx_?numeric, σy_?numeric, τxy_?numeric, θ_?numeric, annotate_?bool,
  onPositiveSideOnly_?bool, limit_?numericStrictPositive, gridLines_?numericPositive,
  {contentSizeW_?numericStrictPositive, contentSizeH_?numericStrictPositive}] := Module[
  {σ1, σ2, σxx, σyy, τxyxy, r, σxxRightArrow, σxxLeftArrow, σyyTopArrow, σyyBottomArrow, τRightArrow,
   τLeftArrow, τTopArrow, τBottomArrow, τ1, τ2, color, textSize = 11, colorShear, σxxRightArrowText,
   σxxLeftArrowText, σyyTopArrowText, σyyBottomArrowText, τRightArrowText, τLeftArrowText,
   τTopArrowText, τBottomArrowText, rotationMatrix, coordinates, from, to, rotatedAxisXText,
   rotatedAxisYText, maxAbsoluteprincipalShearStress, thickness = Thick, eps = 10^-9, θp1, θp2},

  rotationMatrix = RotationMatrix[-θ];
  rotatedAxisXText = Text[Style["x", Italic, textSize], {0.3, 0}.rotationMatrix];
  rotatedAxisYText = Text[Style["y", Italic, textSize], {0, 0.3}.rotationMatrix];

  {{σ1, θp1}, {σ2, θp2}} = principalStresses[σx, σy, τxy];
  {τ1, τ2} = maxAndMinShearStress[σx, σy, τxy];
  maxAbsoluteprincipalShearStress = Max[Abs[{τ1, τ2}]];
  {σxx, σyy, τxyxy} = rotationStress[σx, σy, τxy, θ];

  If[Abs[σ1] > 0,
   {σxx, σyy} = {σxx, σyy} / Abs[σ1](*scale*)
  ];

  If[maxAbsoluteprincipalShearStress > 0,
   τxyxy = τxyxy / maxAbsoluteprincipalShearStress(*scale*)
  ];
  color = Red;
```

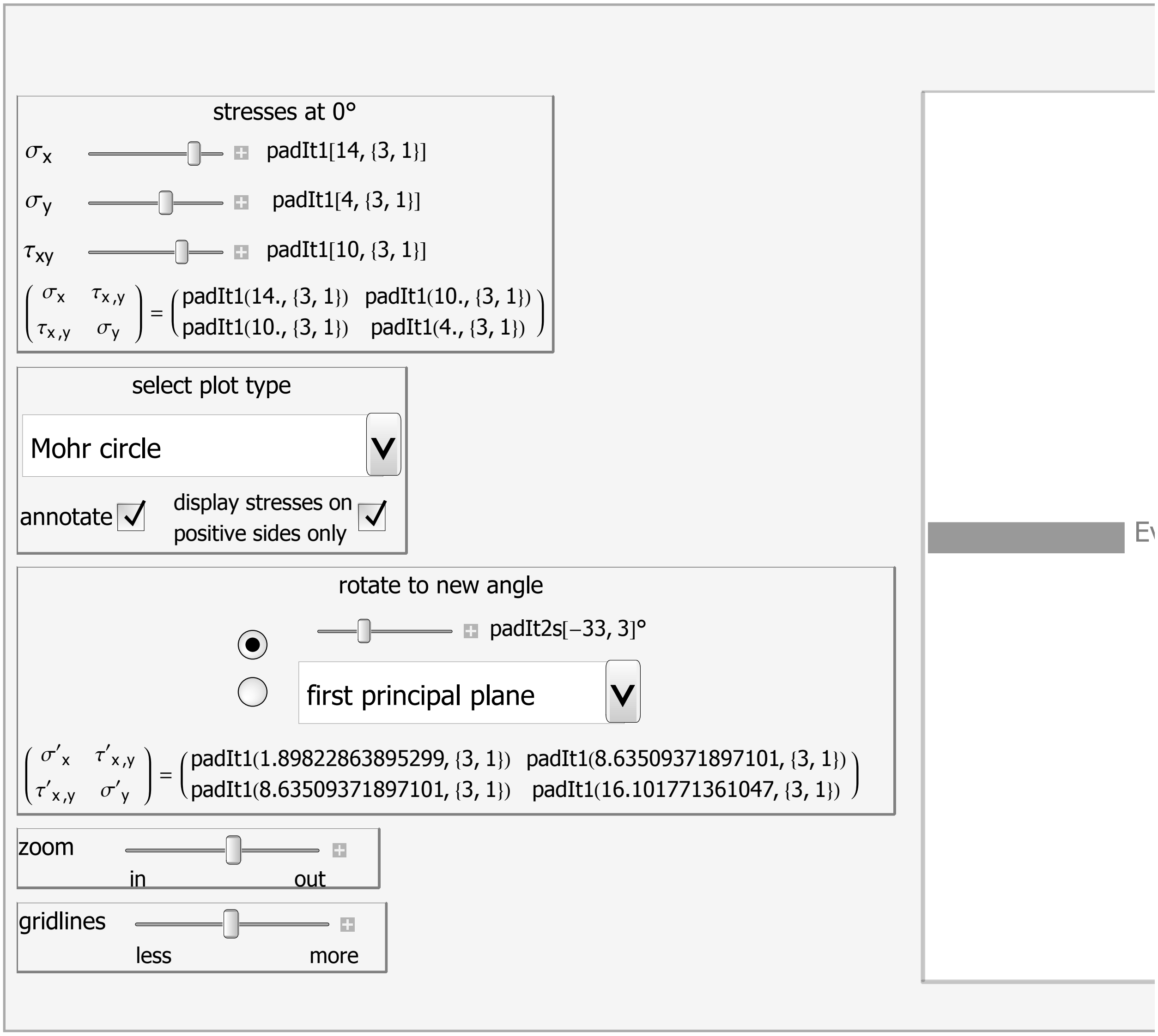
```
colorShear = Blue;
r = {White, EdgeForm[{Thin, Gray}], Rectangle[{-0.5, -0.5}, {0.5, 0.5}]};

(*--------*)
If[σxx ≥ 0,
 from = {0.6, 0};
 to = {0.6 + σxx, 0};
 coordinates = {If[annotate, 0.78, 0.68] + σxx, 0}.rotationMatrix
 ,
 from = {0.6 + Abs@σxx, 0};
 to = {0.6, 0};
 coordinates = {If[annotate, 0.78, 0.68] + Abs@σxx, 0}.rotationMatrix
];

σxxRightArrowText = If[annotate,
   Text[Style[Column[{TraditionalForm[Style["σx"]], padIt1[σxx*Abs[σ1], {3, 1}]},
      Alignment → Center], textSize], coordinates, {0, 0}],
   Text[TraditionalForm[Style["σx", textSize]], coordinates, {0, 0}]
 ];
σxxRightArrow = {thickness, Arrowheads[Medium], color, Arrow[{from, to}, 0]};

(*--------*)
If[σxx ≥ 0,
 from = {-0.6, 0};
 to = {-0.6 - σxx, 0};
 coordinates = {If[annotate, -0.78, -0.68] - σxx, 0}.rotationMatrix
 ,
 from = {-0.6 - Abs@σxx, 0};
 to = {-0.6, 0};
 coordinates = {If[annotate, -0.78, -0.68] - Abs@σxx, 0}.rotationMatrix
];

σxxLeftArrowText = If[annotate,
  Text[Style[Column[{TraditionalForm[Style["σx"]], padIt1[σxx*Abs[σ1], {3, 1}]},
      Alignment → Center], textSize], coordinates],
  Text[TraditionalForm[Style["σx", textSize]], coordinates]
 ];
σxxLeftArrow = {thickness, Arrowheads[Medium], color, Arrow[{from, to}, 0]};

(*--------*)
If[σyy ≥ 0,
 from = {0, 0.6};
 to = {0, 0.6 + σyy};
 coordinates = {0, If[annotate, 0.75, 0.68] + σyy}.rotationMatrix
 ,
 from = {0, 0.6 + Abs@σyy};
 to = {0, 0.6};
 coordinates = {0, If[annotate, 0.75, 0.68] + Abs@σyy}.rotationMatrix
];

σyyTopArrowText = If[annotate,
  Text[Style[Column[{TraditionalForm[Style["σy"]], padIt1[σyy*Abs[σ1], {3, 1}]},
      Alignment → Center], textSize], coordinates],
  Text[TraditionalForm[Style["σy", textSize]], coordinates]
 ];
σyyTopArrow = {thickness, Arrowheads[Medium], color, Arrow[{from, to}, 0]};

(*--------*)
If[σyy ≥ 0,
 from = {0, -0.6};
 to = {0, -0.6 - σyy};
 coordinates = {0, -0.75 - σyy}.rotationMatrix
 ,
 from = {0, -0.6 - Abs@σyy};
```

```
 to = {0, -0.6};
 coordinates = {0, -0.75 - Abs@σyy}.rotationMatrix
];

σyyBottomArrowText = If[annotate,
  Text[Style[Column[{TraditionalForm[Style["σy"]], padIt1[σyy*Abs[σ1], {3, 1}]},
     Alignment → Center], textSize], coordinates],
  Text[TraditionalForm[Style["σy", textSize]], coordinates]
 ];
σyyBottomArrow = {thickness, Arrowheads[Medium], color, Arrow[{from, to}, 0]};

(*--------*)
If[τxyxy ≥ 0,
 from = {0.6, 0.5 - τxyxy};
 to = {0.6, 0.5};
 coordinates = {If[annotate, 0.8, 0.7], 0.45}.rotationMatrix
 ,
 from = {0.6, 0.5};
 to = {0.6, 0.5 - Abs@τxyxy};
 coordinates = {If[annotate, 0.8, 0.7], 0.45}.rotationMatrix
];

τRightArrowText = If[annotate,
  Text[Style[Column[{TraditionalForm[Style["τxy"]], padIt1[
       τxyxy*maxAbsoluteprincipalShearStress, {3, 1}]}, Alignment → Center], textSize], coordinates]
  ,
  Text[TraditionalForm[Style["τxy", textSize]], coordinates]
 ];
τRightArrow = {thickness, Arrowheads[Medium], colorShear, Arrow[{from, to}, 0]};

(*--------*)
If[τxyxy ≥ 0,
 from = {0.5 - τxyxy, 0.6};
 to = {0.5, 0.6};
 coordinates = {0.5, 0.75}.rotationMatrix
 ,
 from = {0.5, 0.6};
 to = {0.5 - Abs@τxyxy, 0.6};
 coordinates = {0.5, 0.75}.rotationMatrix
];

τTopArrowText = Text[TraditionalForm[Style["τyx", textSize]], coordinates];
τTopArrow = {thickness, Arrowheads[Medium], colorShear, Arrow[{from, to}, 0]};

τLeftArrow = {Arrowheads[Medium],
  If[τxyxy ≥ 0,
   {
    thickness, colorShear, Arrow[{{-0.6, -0.5 + τxyxy}, {-0.6, -0.5}}, 0]
   }
   ,
   {
    thickness, colorShear, Arrow[{{-0.6, -0.5}, {-0.6, -0.5 + Abs@τxyxy}}, 0]
   }
  ]};

τBottomArrow = {thickness,
  colorShear,
  Arrowheads[Medium],

  If[τxyxy ≥ 0,
   Arrow[{{-0.5 + τxyxy, -0.6}, {-0.5, -0.6}}, 0]
   ,
   Arrow[{{-0.5, -0.6}, {-0.5 + Abs@τxyxy, -0.6}}, 0]
  ]
```



```
 };

from = -(limit / 40) * 1.9;
to = -from;

Graphics[
 {Rotate[r, θ, {0, 0}],

  If[Abs@σxx > eps,
   {
    Rotate[σxxRightArrow, θ, {0, 0}],
    σxxRightArrowText,
    If[onPositiveSideOnly, Sequence @@ {}, Rotate[σxxLeftArrow, θ, {0, 0}]]
   },
   Sequence @@ {}
  ]
  ,

  If[Abs@σyy > eps,
   {
    Rotate[σyyTopArrow, θ, {0, 0}],
    σyyTopArrowText,
    If[onPositiveSideOnly, Sequence @@ {}, Rotate[σyyBottomArrow, θ, {0, 0}]]
   },
   Sequence @@ {}
  ]
  ,

  If[Abs@τxyxy > eps,
   {
    Rotate[τRightArrow, θ, {0, 0}],
    τRightArrowText,
    Rotate[τTopArrow, θ, {0, 0}],
    If[onPositiveSideOnly, Sequence @@ {},
     {
      Rotate[τLeftArrow, θ, {0, 0}],
      Rotate[τBottomArrow, θ, {0, 0}]
     }
    ]
   },
   Sequence @@ {}
  ]
  ,
  {Gray, Thin, Dashed, Rotate[{Arrowheads[Small], Arrow[{{-0.25, 0}, {0.25, 0}}]}, θ, {0, 0}]},
  {Gray, Thin, Dashed, Rotate[{Arrowheads[Small], Arrow[{{0, -0.25}, {0, 0.25}}]}, θ, {0, 0}]},
  rotatedAxisXText,
  rotatedAxisYText,
  {PointSize[0.01], Point[{0, 0}]}
 }
 ,
 Axes → False,
 PlotRange → {{from, to}, {from , to}},
 ImageSize → {contentSizeW, contentSizeH},
 ImagePadding → {{10, 10}, {10, 10}},
 ImageMargins → 0,
 AspectRatio → Automatic,
 If[gridLines == 0, GridLines → None,
  {
   GridLines →
    {Range[from, to, (to - from) / (gridLines * 20)], Range[from, to, (to - from) / (gridLines * 20)]},
   GridLinesStyle → Directive[Thickness[.001], LightGray]
  }
 ],
 Frame → False
```

```
            ]
          ]
      }
  ]
```



Caption

This Demonstration generates Mohr's circles for plain stress. The input is stress values for σ_x, σ_y, τ_{xy} at the x and y orthogonal faces oriented at zero angle. The Demonstration calculates a Mohr's circle and generates other plots to illustrate how stress changes at different orientations as the angle of the plane is changed using the slider. Planar stress is assumed, therefore stresses in the z direction are assumed to be zero.

Thumbnail

Snapshots

Details (optional)

In plane stress, components σ_z, τ_{zx}, τ_{zy}, τ_{xz}, τ_{yz} vanish and the 3D stress tensor $\begin{pmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{pmatrix}$ reduces to $\begin{pmatrix} \sigma_x & \tau_{xy} \\ \tau_{xy} & \sigma_y \end{pmatrix}$. Assuming σ_x, σ_y, and τ_{xy} are given at 0°, the stresses $\sigma_{x'}$, $\sigma_{y'}$, $\tau_{x'y'}$ at a different angle θ are found from

$$\begin{pmatrix} \sigma_{x'} \\ \sigma_{y'} \\ \tau_{x'y'} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(\sigma_x + \sigma_y) + \frac{1}{2}(\sigma_x - \sigma_y)\cos(2\theta) + \tau_{xy}\sin(2\theta) \\ \frac{1}{2}(\sigma_x + \sigma_y) - \frac{1}{2}(\sigma_x - \sigma_y)\cos(2\theta) - \tau_{xy}\sin(2\theta) \\ -\frac{1}{2}(\sigma_x - \sigma_y)\sin(2\theta) + \tau_{xy}\cos(2\theta) \end{pmatrix}.$$

The angles θ_1 and θ_2 at which the maximum and minimum normal principal stress σ_1 occurs are given by $\tan(2\theta_1) = \frac{2\tau_{xy}}{\sigma_x - \sigma_y}$ and $\theta_2 = \theta_1 + \frac{\pi}{2}$, respectively. The maximum and minimum normal principal stresses are given by $\sigma_{1,2} = \frac{\sigma_x + \sigma_y}{2} \mp \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2}$, where σ_1 is taken as the larger of the two principal stresses in absolute terms. The maximum shear stress is at 45° from the principal plane and is given by $\tau_{max} = \pm(\sigma_1 - \sigma_2)$. At the principal planes the shear stress is always zero. Mohr's circle for plain stress can be viewed from the pulldown menu. You can view polar plots that show how the normal and shear stress change with angle. You can select the angle to view the stresses by using the slider or select specific planes using the pulldown menu. All units are assumed to be in SI.

References

[1] A. C. Ugural, S. K. Fenster, *Advanced Strength and Applied Elasticity*, New York: Elsevier, 1987.

[2] REA's *Problem Solver's Strength of Materials & Mechanics of Solids,* New Jersey: Research and Education Association, 1996.

[3] Irving H. Shames, *Mechanics of Deformable Solids*, Prentice-Hall, Inc. N.J. 1964.

Control Suggestions (optional)

- [x] Resize Images
- [x] Rotate and Zoom in 3D
- [x] Drag Locators
- [ ] Create and Delete Locators
- [x] Slider Zoom
- [x] Gamepad Controls
- [x] Automatic Animation
- [ ] Bookmark Animation

Search Terms (optional)

plain stress

principal stress

normal stress

shear stress

Mohr's circle

Related Links (optional)

analysis of stress

Authoring Information

Contributed by: Nasser M. Abbasi